

VHDL Testbenches and Verification

4 Days: 50% lecture, 50 % Lab

Advanced Level

Overview

Students will gain in-depth knowledge of advanced VHDL coding styles and methodologies to verify digital hardware (ASICs and FPGAs). Test topics range from the creation of simple testbenches through the creation of a transaction-based, system-level, self-checking test testbench. Transaction-based testbenches simplify test development and significantly reduce the time it takes to write tests.

This course starts by showing how to create a simple test environment and progressively adds to it. Along the way students learn about subprogram usage, TEXTIO, modeling issues, transaction-based tests, data structures (linked-lists, scoreboards, memories), algorithmic and random test generation, protocol checking, result checking, and handshaking methods. Once putting the basics in place, the course starts looking at the system-level testbenches. A system-level testbench replicates the environment of the design under test. Each interface in the system is implemented by a behavioral model. Lecture and laboratory examples progressively add timing checking, protocol checking, data input (source), data result checking, and data synchronization facilities to the models.

Intended Audience

VHDL Testbenches and Verification is recommended for experienced VHDL designers who need in-depth knowledge on verification coding techniques.

Course Objective

Upon completion of this course, students will be able to:

- Create a transaction-based, system-level, self-checking test environment
- Use textio to read files and write useful messages
- Write bus functional models for testbenches
- Use subprograms to abstract interface actions (CpuRead, CpuWrite)
- Understand how VHDL executes a model
- Plan the test process from subblock to system-level tests
- Know how to create a test plan
- Use VHDL configurations to simplify and increase the effectiveness of tests
- Model RAM

VHDL Testbenches and Verification

Course Outline

Day 1, Module TB1

Testbench Overview
Basic Testbenches
Transactions and Subprograms
Modeling for Verification
VHDL IO

Day 2, Module TB2

Lab Review: Testing w/ subprograms
Transaction-Based BFM
Elements of a Transaction-Based BFM
Execution and Timing

Day 3, Module TB3

Lab Review: UartTx BFM
Creating Tests
Data Structures for Verification
Constrained Random Tests

Day 4, Module TB4

Test Plans
Configurations and Simulation
Management
Modeling RAM
Transaction-Based BFM Part 2

Prerequisites

Students taking this course should have working knowledge of digital circuits and prior exposure to VHDL through experience or the course:

Comprehensive VHDL Introduction - 4 days

Other Recommended Courses

Students may also be interested in the following companion course:

VHDL Coding for Synthesis - 4 days

Customization

All of our courses can be customized to meet your specific needs. Either see our website or contact us for details.

Training Approach

This hands-on, how-to course is taught by experienced verification engineers using a computer driven projector. We prefer and encourage student and instructor interaction. Questions are welcome. Bring problematic code.

Contact

To schedule a class or for more information, contact:

Jim Lewis
Director of Training
(800) 505-8435 / (800) 505-VHDL
jim@SynthWorks.com
<http://www.SynthWorks.com>

Learn VHDL from a designer's perspective with SynthWorks.